

Ahmed Alzubairi

Open Source Development

Professor Timothy Paine

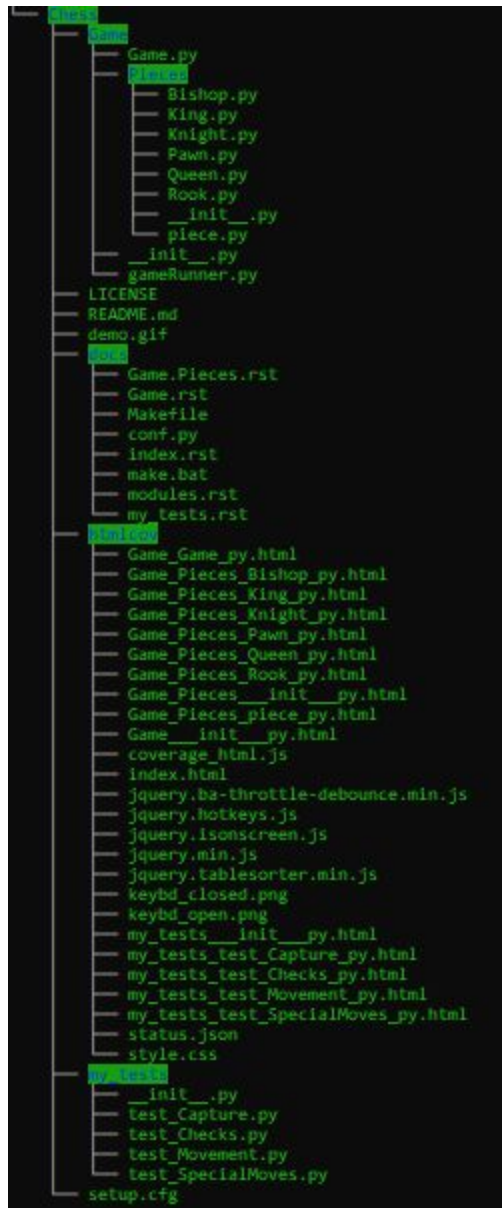
22 December 2020

Final Report

Chess Backend Github link: <https://github.com/AhmedAlzubairi1/Chess>

Not to say that I am that much of a great player, but I really enjoy playing it with my friends and father. It was one of the few ways I bonded with my dad, so the game was a big part of my childhood in that regard. Since my intro cs course I've wanted to work on a chess game, but never had the time. Since we were tasked with working on our own open source project, I felt that the Open Source Development project would be a perfect opportunity for me to develop my chess backend. Since I was most comfortable with python, I chose to write the project in python.

In a nutshell, I worked on creating a chess backend that had all the functionality of a chess game. The idea is that other developers in the open source community can add more features to it and add their own frontend implementations of my backend. Even though I wanted to work on just the backend, I implemented a basic frontend implementation to show developers an example of my work in action, thus I decided to implement my Chess game through terminal console output. The project fully works and has all the necessary chess rules and functionality that one would expect of Chess. My dream end state is that developers would create their own frontend implementations and make a pull request to add their version into the Game directory. An ideal goal would be that the Game directory would host various frontend implementations to play Chess. Currently my terminal frontend is in the Game.py file and can be run through the gameRunner.py file in the Game directory. This is the codebase layout of my project:



License Badge and Tags	shields.io	Used shields.io to create the needed badges.
Create Issues	github	Added issues for various issues I wanted to solve like add testing, add docs, and also released the v1
Testing	pytest	I ran pytest in the my_tests/ directory to run all my tests.

		The tests are to make sure the rules for the game work as well as to check that all the pieces can successfully move and capture pieces. Pytest was also used in the CI to run the tests during every pull request. Travis was used to require the tests to pass.
CI	travis	It is set up to run on all branches and pull requests. It installs the needed dependencies, builds, lints, runs testing, and submits coverage.
Code Coverage	codecov	Travis runs the coverage report for codecov and submits the coverage reports to Codecov where the reports as well as stats and analysis of them is held.
Lint	flake8	I run flake8 with ignoring the E501, W504, E712 warnings. E501 was ignored because I wanted some of the lines to be long (very few). W504 and E712 warnings were fixed but flake8 still gave me the warnings for some reason, so I ignored them. Changes are seen in the setup.cfg file.
Documentation	Sphinx & Readthedocs	Hosted the documentation through Readthedocs and set up API auto-documentation for all my methods. I used Sphinx with the sphinx_rtd_theme theme.
Website/Web Presence	github pages	Set up a personal webpage using github pages with links to my project. Link to website is found on github repo.

Release Notes & Release	github	Released my version with release notes through github. First release is labeled v0.0.1
Examples/Demos	Github	Added a gif to demonstrate how to run the terminal frontend. It can be seen in the repo's Readme.
Second Release	github	Second release is labeled v1.0.0

Before this class I didn't really know what open source development meant or was. I knew it used Github, but that's pretty much it. Before the only ways I interacted with Github was through basic git commands I learned in Advanced Programming. With this class, I now know what all those tabs on github are used for. I know how to contribute to open source projects as well as learning a lot of valuable tools like travis. I also learned how to do more robust testing and work with continuous integration. Pretty much everything in this class was new to me and I am happy that I learned them because this is by far one of my top 3 courses I've taken at Columbia. I am happy I took the course because it is great exposure to the many tools I will most likely run across in my software engineering career.

One thing that was annoying about the project was the bugs. I've messaged Professor Paine numerous times on Gitter just to realise the issues were due to some trivial silly thing. I will never forget the time where I spent hours working on getting Sphinx auto documentation to work just to realize that the issue was because my machine was running python 2 instead of 3. I didn't even know I had python 2 to begin with. It was embarrassing, but also rewarding in a way. Shows how the little bugs can often take the longest to tackle.